IDA

INSTITUTE FOR DEFENSE ANALYSES

A Linear Programming Approach to Complex Games: An Application to Nuclear Exchange Models

I. C. Oelrich, Project Leader

August 2002

Approved for public release; distribution unlimited.

IDA Document D-2743

Log : H 02-001523

This work was conducted under IDA's Independent Research Program, CRP 1082. The views expressed in this report are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

INSTITUTE FOR DEFENSE ANALYSES

IDA Document D-2743

A Linear Programming Approach to Complex Games: An Application to Nuclear Exchange Models

I. C. Oelrich, Project Leader
C. D. Farmer

PREFACE

This paper reports an analytical solution to a two-sided nuclear exchange using simple linear programs. This solution is proposed as an alternative to MESA, a model developed at Los Alamos National Laboratory that uses an optimizing search algorithm. Like the MESA model, the exchange is cast in terms of game theory, using linear approximations and an optimal allocation defined by a user-specified objective function. Solutions are better using linear programs instead of optimizing searches, and the solutions are many times faster.

Catherine D. Farmer thanks Mr. Philip L. Major, IDA Vice President for Planning & Evaluation, for support from the IDA summer intern program. The authors also thank Dr. Phillip Gould for reviewing this paper and providing helpful comments.

(This page is intentionally blank.)

CONTENTS

Appendixes

(This page is intentionally blank.)

LIST OF FIGURES

(This page is intentionally blank.)

LIST OF TABLES

(This page is intentionally blank.)

EXECUTIVE SUMMARY

The Defense Threat Reduction Agency's (DTRA) Advanced Systems and Concepts Office (ASCO) tasked IDA to evaluate MESA, a nuclear exchange model developed at Los Alamos National Laboratory (LANL). MESA, the Multiple Exchange of Strategic Arsenals, optimizes the allocation of weapons against targets. However, in the resulting IDA report, it was noted that MESA produces "almost pure" solutions. That is, warheads do not exhaust one weapon-target combination before being allocated to another weapon-target combination. However, if warheads are allocated to the "pure" solution through hand calculation, a better objective function value results.

Therefore, an alternative solution was sought. We approached the development of this new solution with the assumption that, if the linear approximations in MESA are acceptable, the whole problem should be linear. Thus, once a good weapon-target combination is found, the derivative of the objective function should remain constant until either weapons or targets are exhausted. By this method, the optimal allocation of weapons should be a "pure" solution.

With the linear approximations already contained in MESA, we were able to find a completely linear solution. The parameter space was explored systematically and thoroughly. The solutions appear to be as good as, or better than, MESA's in all cases tested. They are analytical and do not depend on initial search conditions as MESA's do. And the solutions are produced in computer runs that are two orders of magnitude faster.

(This page is intentionally blank.)

I. A LINEAR PROGRAMMING APPROACH TO COMPLEX GAMES: AN APPLICATION TO NUCLEAR EXCHANGE MODELS

A. INTRODUCTION

The Advanced Systems and Concepts Office (ASCO) of the Defense Threat Reduction Agency (DTRA) tasked the Institute for Defense Analyses (IDA) to review MESA (Multiple Exchange of Strategic Arsenals), a computer program developed at the Los Alamos National Laboratory (LANL). MESA is a nuclear exchange model that can determine a country's optimal allocation of warheads given the country's arsenal and war goals as well as the opposition's arsenal and war goals.

The IDA review of MESA[1] noted that under special conditions the optimal weapon allocation produced by MESA results in an almost "pure" solution. That is, any specific weapon tends to be allocated almost entirely against one particular target type. Hand calculation shows that the actual "pure" solution (allocating warheads in a certain weapon-target combination until either all warheads or all targets have been exhausted) results in a better objective function value than the almost pure solutions typically found by MESA.

The motivation for this study was to investigate the origins of these "almost pure" results and to explore an alternative approach to a solution. This report gives an overview of the MESA model and demonstrates an alternative approach that finds better solutions and cuts run time dramatically.

---

[1] *Evaluation of MESA, a Nuclear Exchange Model, to Explore War Goals, Arsenals, and Stability*, IDA Paper P-3659, November, 2001, pp. A1 – A5.

B. OVERVIEW OF THE MESA MODEL

The MESA model consists of a set of three nested optimizations.[2] The outer optimization gives the percentage of weapons to be used in the current exchange. Unfortunately, the long run times of the inner optimizations have rendered the automation of this optimization impractical, preventing MESA from solving multiple exchanges of arsenals. Therefore, the user performs this step. In the middle nested optimization, weapons are allocated between counterforce and countervalue targets. In general, weapons are force targets and everything else is a value target. In the innermost box of the optimization, warheads are allocated to each target type within the counterforce and countervalue groupings.

MESA requires the user to input the damage expectancy (DE) and the required damage (RD). DE is an indicator of the goodness of a weapon. There is a damage expectancy assigned for each weapon on Side One against each target on Side Two, for both force and value targets, and similarly for the weapons on Side Two in retaliation against Side One targets. The "required" damage is the damage required by the attacking side for the attack to be considered a success. In all cases considered here, RD is set to be the same for all weapons in an attack. Our alternative approach to solving the optimization utilizes the more common P_k, the "probability of kill," which is a measure of the likelihood that the warhead will destroy a given target. As with the DE, each weapon-target combination has a P_k. The relationship between these values is:

$$P_k = \log(1 - DE) / \log(1 - RD)$$

The innermost nested optimization in MESA uses a linear program to determine the best allocation of weapons that have been assigned to either counterforce or countervalue. Certain approximations are needed to make the payoff functions linear. For example, the effects of multiple weapons used against a target are additive. That is, two weapons each with a P_k of 0.4 have a total P_k of 0.8 rather than 0.64. MESA constrains the cumulative P_k such that it cannot be greater than one. To mimic fratricide

---

2 Ibid., pp. 2.

constraints, MESA allows no more than two warheads against any given target. Additionally, the number of destroyed targets cannot exceed the number of targets present. That is to say, P_k times the number of allocated weapons must be less than the number of targets present.

MESA has available several objective functions for which the user can find an optimal allocation of weapons. In this study, the simplest, and original, objective function was used in all cases. MESA attempts to minimize this objective function, henceforth "F", which is defined as follows:

$$F = D_S + \lambda (D_G - D_I)$$

where D_S = Damage Suffered, D_G = Damage Goal, and D_I = Damage Inflicted. The damage goal, in all cases in this study, equals the total number of value targets on the opposing side. The values for damage suffered and damage inflicted are found by multiplying the weapon allocation against value targets by the appropriate P_ks. λ is a value assigned by each side to indicate its willingness to accept damage as compared to inflicting damage. That is, a λ of 0.2 reflects an indifference between destroying five enemy targets or preserving one friendly target.

To confirm that we completely understood the outputs produced by MESA, we worked through a process of reproducing these outputs by hand. Aware of the basic constraints and inputs, the first approach was to ensure that calculating P_k as defined above and multiplying this value by the number of warheads allocated in the MESA output would yield the F values calculated by MESA. We were able to do this exactly.

Once we were able to reproduce the MESA results by hand, we were able to make small adjustments in the MESA results, converting almost "pure" solutions to truly pure solutions. We were able to show that the pure solutions were as good as or better than the MESA-produced solutions, that is, they generated a value of F that was equal to or lower than the MESA value. With the simple arsenals[3] used here, MESA runs in 15 to 45

[3] In the simple arsenal, each side has five weapon types and two value target types.

minutes on an 800 MHz Pentium III computer. The fact that we could produce improved results by hand in a couple of minutes suggested that something was amiss.

C. A NEW APPROACH FOR SOLVING THE MESA PROBLEM

1. Hypothesis

After confirming our understanding of P_k, we faced the problem of reproducing the allocation of weapons found in MESA (or finding better allocations than MESA's). We developed a hypothesis to account for the almost "pure" solutions that MESA tended to find. This hypothesis is based on the premise that the whole problem should be linear, because the linear objective function is combined with the linearized payout functions. Further, we speculated that once an optimal weapon-target combination is found, the derivative of F remains constant until either targets or weapons are exhausted. In this case, the optimal allocation will result in a "pure" solution. However, MESA produces almost "pure" solutions only because of limitations of the simulated annealing optimization, whereby weapon allocations are tested until the difference between consecutive allocation tests is so small that the allocations are considered to be "good enough." Therefore, we set out to see whether, *with the approximations already contained in MESA*, we could find a completely linear solution.

2. Examination

To solve this problem, we started at the end and worked backward through the process. We first looked at the case of the retaliator firing at the first striker, but with no degradation of his arsenal. In this manner, we can find the optimal allocation of Side Two's[4] weapons when his arsenal is untouched. Because he is the second striker, he cannot improve his objective function by allocating weapons to counterforce targets, so all warheads should be aimed at Side One's value targets. The allocation of Side Two's weapons was calculated by looking at a simple payout matrix where Side Two attempts

---

[4] MESA calls the two sides "Side One" and "Side Two" so that Side One could be the second striker. However, we always refer to the first striker as "Side One" and the second striker as "Side Two."

to gain the maximum payout from the matrix (entries of the matrix are ultimately the P_k of each weapon against the given target) times the target value (in all cases in this study, each target has a value of one).

The naïve solution to such a matrix problem is to pick the best weapon-target combination, then the second best combination from the remaining combinations, and so on. Take the matrix in Table 1, assuming there exists one of each type of weapon and one of each type of target and a maximum allocation of one weapon per target:

Table 1. A Simple Side-Two Payout Matrix

Weapon	Target	
	1	2
1	0.9	0.7
2	0.3	0.8

In this example, the naïve solution works well. The best weapon-target combination is weapon one-target one. Therefore, weapon one "picks" target one and weapon two is left to fire against target two. The payout is 1.7 (0.9 + 0.8), which is the best payout possible with the given weapons and targets.

However, the solution is sometimes not as straightforward as it is in this case. There are certainly cases where both weapons reach their optimal payout by firing at target one. If this occurs, which weapon gets priority? Table 2 illustrates a matrix that poses this problem, taking again one of each target type and one of each weapon type with no more than one weapon allocated to each target.

Table 2. A Subtle Side-Two Payout Matrix

Weapon	Target	
	1	2
1	0.9	0.7
2	0.8	0.3

Because there are only two possible allocations of the weapons, the reader can easily see that the best solution is for weapon one to fire against target two while weapon two fires at its most lucrative target, target one. The logic of this choice is that the attacker "loses less" by moving weapon one to its second best choice than he would lose by moving weapon two (0.2 compared to 0.5). However, the solution to such a problem cannot be solved by hand when there are a dozen weapons and equally many targets, with different numbers of each type of weapon and target.

3. ONEGULP

The solution was found by writing ONEGULP, Optimized Nuclear Exchange Games Using Linear Programming. ONEGULP is a linear program in AMPL, a mathematical programming language developed at Bell Labs. Relying on the same main constraint used in MESA of P_k being linear, this program can solve the optimal payout for a 17x17 matrix in a few seconds. The model file and an example data file can be found in Appendix A. By taking cases with all possible relationships between the number of targets and the number of weapons, we became convinced that the program would find the optimal allocation under all conditions.

The model in Appendix A can solve the optimal allocation of weapons for one side against the other with no retaliation. However, this study looks at a two-sided exchange: Side One fires on Side Two, and Side Two is able to retaliate with whatever weapons it has remaining. The goal is to find the optimal allocation for Side One given the war goals, arsenal, and P_ks of both sides. Therefore, the final program must determine how many weapons to allocate to Side Two's targets, which includes both value targets, which contribute directly to reducing Side One's objective function, and force targets, which contribute indirectly to Side One's utility by potentially increasing the damage suffered by Side One if they are not attacked.

4. The Two-Sided Exchange

Taking the two-sided exchange into account proved to be more difficult. Side One must recognize that Side Two's optimal allocation will change as its number of weapons changes. For example, in Table 2 above, if Side Two has 100 warheads of each type of weapon, it will follow the optimal allocation found in Appendix A. However, what happens if Side One destroys one of Side Two's warheads in an attack? The answer depends on which weapon is destroyed. If weapon one is destroyed, then the optimal allocation will not change—the 100 warheads of weapon two will be allocated to target one, and the remaining 99 warheads of weapon one will be allocated to target two. However, if a warhead in the weapon two arsenal is destroyed, the optimal allocation will shift. The 99 remaining warheads of weapon two will be allocated to target one, but weapon one's allocation will change; one warhead will shift and be fired at target one while the remaining 99 continue to be fired at target two.[5]

Analyzing the target choice by comparing the objective function outputs provides a clearer picture. With 200 weapons, Side Two can obtain a maximum payout of 150. With one less weapon, that payout can be reduced to either 149.3 or 149.4. Side One wants to minimize the possible damage done by Side Two, so it shoots at weapon one to reduce the payout to 149.3.[6] When choosing the Side Two weapons upon which it should fire, Side One must find the weapon that creates the largest decrease in payout. In order to find the desired force target, Side One must look at the weapon with the highest P_k for all targets, not just the weapon with the highest P_k for the optimal allocation of weapons. The weapon that fulfills this requirement is also the weapon that will result in the largest decrease in F. This weapon will be fired upon until either it or the weapon shooting at it is exhausted. At this point, the process of finding the weapon resulting in the largest decrease in F is repeated. In this manner, the derivative of F remains constant until the weapon-target combination is depleted. When the weapon-target combination changes, the derivative of F will also change, but will again remain constant until the next shift in allocation.

---

5 Note: For simplicity in discussion, the assumption of one warhead per target is maintained, this constraint will be lifted in the final ONEGULP model.

6 In this part of the discussion, the assumption is made that all Side One weapons have the same P_k against all Side Two weapons. Following this logic, it will be easy to incorporate the Side One P_k s in the final payout matrix.

Incorporating this logic into ONEGULP required appropriately defining the payout matrix described earlier. Within the payout matrix for Side One, there are two components: one of weapons against force targets (weapons) and one of weapons against value targets. The payout for a weapon against a value target is simply defined as λ times the P_k of the weapon-target combination. The payout for a weapon against force target can be calculated by multiplying the P_k of that weapon-weapon combination times the Δ F(1) of the Side Two weapon, where Δ F(1) is the change in the Side One objective function that results from destroying that weapon. Δ F(1) is found for each Side Two weapon with the following formula:

$$- \Delta F(1) = \max (P_{k2}) - (\max (P_{kC}) - P_{kA})$$

where max (P_{k2}) equals the maximum P_k of the Side Two warhead against any Side One value targets; max (P_{kC})[7] equals the maximum P_k of any Side Two weapons against the value target specified by max (P_{k2}); and P_{kA} equals the P_k of the optimal weapon-target combination for the Side Two weapon specified by max (P_{kC}). Appendix B demonstrates an example of the calculation of the Side One payout matrix. Each value in the matrix gives the value that will be *subtracted* from Side One's objective function where the starting objective function value is defined as the objective function Side One would have if it did not fire any warheads, but was fired upon. Because Side One wants to minimize its objective function, it will find the allocation that maximizes the payout matrix.

With this payout matrix as an input, a program similar to the one discussed earlier was written to find the optimal allocation of warheads by Side One (the code for this program can be found in Appendix C). Because we could not mathematically *prove* that this solution always finds the minimum objective function value, we explored the space of all feasible conditions to test our model. These tests include all possible combinations between the number of weapons and the number of targets for each side. Then, each of these combinations was tested with higher and lower λs and increasing the number of warheads per platform for Side Two. A total of over 250 cases were tested. While the mathematical proof eluded us, this thorough testing procedure provided compelling evidence supporting our original hypothesis.

D. DISCUSSION OF RESULTS

To compare the results of ONEGULP and MESA, we looked at the values each produced for F(1), the Side One objective function, and F(2), the Side Two objective function. In 87 percent of the test cases, the MESA and ONEGULP values for F(1) were within 3 percent of each other. These results fall into four categories:

1. F(1) and F(2) in ONEGULP are smaller than F(1) and F(2) in MESA
2. F(1) in ONEGULP is smaller than F(1) in MESA while F(2) in ONEGULP is larger than F(2) in MESA
3. F(1) in ONEGULP is larger than F(1) in MESA while F(2) in ONEGULP is smaller than F(2) in MESA
4. F(1) and F(2) in ONEGULP are larger than F(1) and F(2) in MESA.

This discussion of the results will take each of these cases separately to show why the differences in the results of MESA and ONEGULP exist.

1. Both Values Smaller in ONEGULP Than in MESA

When the F(1) value in ONEGULP is smaller than the F(2) value in MESA, as it is in the first two categories, ONEGULP is simply coming up with a better allocation of Side One weapons than MESA. As discussed previously, when MESA allocates weapons to an almost "pure" solution; the "pure" allocation of weapons results in an F value that is equal to or lower than (and therefore better than) the F value produced in MESA with the almost "pure" allocation. Therefore, it is not surprising that we see a

---

[7] The P_{kC} notation is used to show that we are finding the max (P_k) in a given column (as defined by (P_{k2}). The P_{kA} notation is used to show that we are looking for the P_k of the optimal weapon allocation in a given row.

number of cases where the value of F(1) produced in ONEGULP is lower than the value of F(1) produced by MESA.

In the case where both F(1) and F(2) in ONEGULP are lower than the corresponding values in MESA, the change from an almost "pure" allocation to a "pure" allocation of weapons results in a slight change in the number of weapons remaining for Side Two. This change in the number of warheads remaining leads to a corresponding change in the allocation of weapons on Side Two, thus resulting in a slight change in F(2). In all cases tested that fell into this category, the decrease in F(2) from MESA to ONEGULP was never more than 2 percent. This slight decrease is explained by the correction of warhead allocation after the correction of the almost "pure" solution.

2. ONEGULP's F(1) Value Smaller and F(2) Value Larger Than MESA's

The second category defined above, where the value of F(1) in ONEGULP is smaller than in MESA and the value of F(2) in ONEGULP is larger than in MESA, typically occurs when there can be a number of weapon allocations that produce the same minimum F(1). However, each of these allocations leaves Side Two in a different retaliatory position, which can cause dramatic differences in F(2) values. Table 3 gives an example of the variance in F(2) obtained through different weapon allocations of the same arsenal. The seed number refers to an input in MESA that just sets an arbitrary starting point in the allocation space where the algorithm will begin its search. While there is no correlation between the seed number and the values in the table, there is a correlation between the weapon allocation of Side One and the value of F(1).

While F(1) varies only from a minimum of 1204.75 to a maximum of 1216.33, F(2) values range from 1396.79 to 1855.04 with MESA and extends to 2153.95 with the ONEGULP output. This variation in F(2) can be explained by looking at the warhead allocation of Side One. As the ratio of counterforce to countervalue warheads decreases, the value of F(2) increases because losing a value target causes a larger increase in Side Two's objective function than losing the indirect value of a force target, which equals Side Two's λ times that weapon's P_k against Side One targets. In this case, ONEGULP chooses the warhead allocation by Side One that not only minimizes F(1), but maximizes

F(2). Because Side One is on an indifference curve, he is able to make Side Two worse off without changing his own position.

Table 3. Variance in F(2) Along an Indifference Curve[a]

Seed Number	F(1) Value	F(2) Value	Counterforce WH fired[b]	Countervalue WH Fired	CF/CV Ratio
490976	1209.07	1396.79	1983.90	1016.20	1.95
102409	1211.09	1449.84	1914.30	1085.70	1.76
248657	1212.91	1459.42	1901.20	1098.80	1.73
748929	1208.88	1572.38	1756.50	1243.60	1.41
949622	1213.21	1643.82	1662.30	1337.80	1.24
27853	1211.96	1649.98	1654.80	1345.30	1.23
918893	1216.33	1658.10	1642.60	1357.50	1.21
436763	1214.07	1710.97	1575.00	1425.10	1.11
662384	1211.98	1745.54	1531.00	1469.10	1.04
645890	1207.98	1761.62	1511.70	1488.40	1.02
156075	1204.75	1770.07	1502.00	1498.00	1.00
500388	1214.41	1855.04	1388.20	1611.90	0.86
AMPL	1204.76	2153.95	1004.70	1995.30	0.50

[a] Conditions used for this test: Side One: 3000 Weapons, 2500 Value Targets, λ = 0.3; Side Two: 3000 Weapons, 2500 Value Targets, λ = 0.3

[b] Warhead allocation for side one.

3. ONEGULP's F(1) Value Larger and F(2) Value Smaller Than MESA's

The third category of results, where the F(1) value produced by ONEGULP is larger than the value produced in MESA and the F(2) value is smaller, occurred most often when λ was small. The weapon allocation of Side Two explains the reason we see this result. In the MESA model for each of these cases, Side Two was not optimizing his objective function, thereby allowing Side One to do "better than his best" possible scenario, effectively lowering the Side One objective function and increasing Side Two's objective function.

The weapons exchange in these trials of MESA and ONEGULP is a two-sided exchange. Side One fires all weapons that will help its objective function, after which Side Two fires all remaining weapons that will help its objective function. Side One does not have the opportunity for retaliation in a third strike. Because weapons offer no intrinsic value in the objective function, Side Two is essentially wasting any weapons it fires at Side One force targets. If counterforce weapons fired by Side Two are allocated to value targets, D_I will increase for Side Two while D_S increases for Side One, thereby causing F(1) to go up while F(2) goes down.

For all the cases in this category, the largest increase from F(1) in MESA to F(1) in ONEGULP was less than 3 percent.[8] Figure 1 illustrates the relationship between the number of weapons Side Two fires at force targets and the value of F(1) for one arsenal with varying seed number.[9] The ONEGULP value can be found in the upper left side of the chart, with zero warheads fired at counterforce targets by Side Two. In contrast, MESA allocates a number of Side Two weapons against counterforce targets. One can see that firing more warheads at counterforce targets instead of countervalue targets decreases Side One's D_S without increasing Side Two's D_I. Thus, this allocation provides an explanation for the cases where the F(1) value for MESA is reported lower than the F(1) value for ONEGULP and the F(2) value reported is higher.

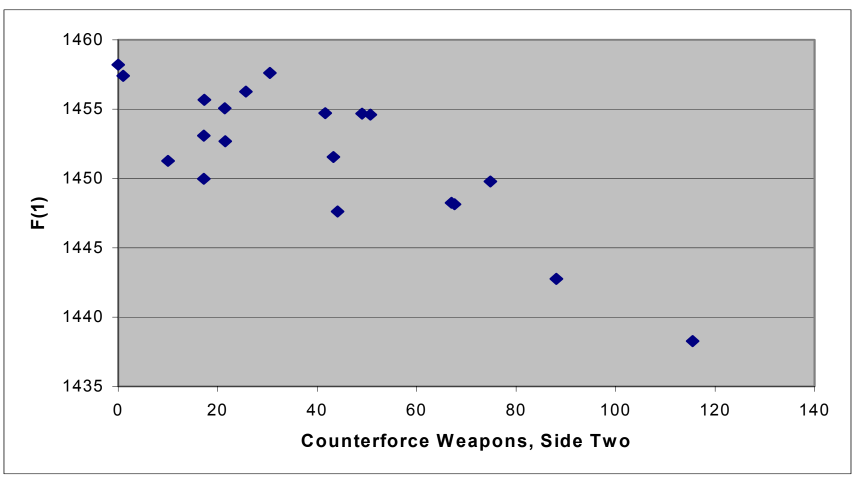

Figure 1. F(1) Versus Side Two Weapons Fired Against Force Targets

---

8 All trials were initially run and percentages were calculated with a random seed of 748929.

9 Conditions in this test: Side One: 2500 Weapons, 2000 Value Targets, $\lambda = 0.3$; Side Two: 3000 weapons, 3000 Value Targets, $\lambda = 0.3$

4. Both Values Larger in ONEGULP Than in MESA

The final category of results is the case where both F(1) and F(2) are larger in ONEGULP than the corresponding outputs in MESA. As with the previous case, these outputs are the results of a misallocation of weapons. It is important to note that both MESA and ONEGULP assume that an exchange of weapons will take place and that weapons will be fired at the opposition so as to inflict damage and avoid damage, in a relationship determined by that country's λ. Because of the definition of the objective function, it is very unlikely that a country will end up with a lower F value than its beginning F value. If both sides fire all of their weapons into space, thus inflicting and suffering no damage, their F value will be $\lambda * D_G$. Once the weapons exchange begins, each side is trying to minimize $D_S - \lambda * D_I$. Because it is unlikely that a country will have a λ near or above one, we can see that it will take a large amount of damage inflicted to cancel out the damage suffered. This is to say that both programs try to minimize down from the F resulting from an unanswered attack, but F values less than this "minimum" are physically, if not logically, allowed. MESA (and therefore ONEGULP) assumes that the exchange will take place. Therefore, the results in this last category are a misallocation of weapons. Because both sides assume that the other will be inflicting damage, they should not waste their own weapons by shooting them in a manner such that they will not optimally inflict (or avoid) damage. In these cases, however, MESA is misallocating both Side One and Side Two weapons.

It is worthwhile to note that the relationship between ONEGULP and MESA outputs varies as the seed number in MESA changes. Table 4 shows an example where, with some seed numbers, the results fall into category three, while other seed numbers shift the results to the fourth category. An advantage of ONEGULP is that the optimal allocation is found each time, not just a "close enough" answer. Therefore, the ONEGULP results are the same on each run. This continuity allows the user to understand the process by which ONEGULP solves an allocation.

Table 4. Variation in Results with Varying Seed Number[a]

Seed Number	F(1)	F(2)
748929	1767.68	1962.20
288272	1807.08	2155.52
817314	1808.66	1786.77
659965	1808.91	1900.80
ONEGULP	1809.74	2557.46
459790	1812.36	1824.17
58435	1815.12	2370.60
353553	1815.58	2122.11

[a] Conditions for this test: Side One: 2500 Weapons, 4000 Targets, λ = 0.3; Side Two: 3000 Weapons, 4000 Targets, λ = 0.3

In addition to aiding bidding strategy, ONEGULP can be applied to the control of networks, such as airfare or spectrum networks. Taking airfare as an example, there are many paths among various cities with airports. However, not all airlines operate along all paths. Instead, they choose which paths will make the most money for them. Therefore, less traveled paths end up with fewer, more expensive, flights than the more traveled flights. Airlines must decide whether they should compete in the smaller markets where they can force competition, and force less profits on the part of their competitors, or if they should stick to the more popular routes. The airline must decide the optimal allocation of its assets to be the most competitive.

As these examples show, ONEGULP has applications that extend past nuclear exchanges. It can be used to analyze many kinds of complex games.

II. CONCLUSION

The ONEGULP model, based on the premise that the weapon allocation problem should be completely linear, is able to produce answers that are as good as, or better than the outputs in MESA. Additionally, ONEGULP operates at speeds up to three orders of magnitude faster than the MESA model. The MESA model generates a run time of 10 minutes to one hour for simple arsenals. The range in run time is caused by a large output file in MESA. MESA outputs are displayed in a series of about 30 files. One of these files can be as large as 100,000 kB. As MESA is run repeatedly, the size of the output files can increase MESA's run time. We ran ONEGULP through the AMPL website at http://www.ampl.com. Through this website, a user can enter the coding, and the linear program will be solved in a matter of seconds. This time measurement was our comparison for the ONEGULP program.

The increased efficiency of the ONEGULP model over the MESA model can be quite valuable. With the decreased run time, the multiple exchange of arsenals could be added without causing the program to be too time consuming. Additional work could also be done to lift some of the constraints, namely the linearity of P_k. While this constraint was helpful in making the allocation solvable by a linear program, it colors the allocation by not subtracting the cases when both warheads are successful against a given target.

Additionally, the decreased run time can allow for a wider application of the concept. There are many examples of hostile bidding and the control of networks where the same strategy is applied. For example, professional sports teams choose the players they want to sign and trade. When making these decisions, the team must consider not only how much a player is worth to the team, but also how much worth is being withheld from other teams by keeping the player on the team's roster. Similar examples can be seen with campaign expenditures and in corporate mergers and acquisitions.

In addition to aiding bidding strategy, ONEGULP can be applied to the control of networks, such as airfare or spectrum networks. Taking airfare as an example, there are many paths among various cities with airports. However, not all airlines operate along all paths. Instead, they choose which paths will make the most money for them. Therefore, less traveled paths end up with fewer, more expensive, flights than the more traveled flights. Airlines must decide whether they should compete in the smaller markets where they can force competition, and force less profits on the part of their competitors, or if they should stick to the more popular routes. The airline must decide the optimal allocation of its assets to be the most competitive.

As these examples show, ONEGULP has applications that extend past nuclear exchanges. It can be used to analyze many kinds of complex games.

Appendix A

MODEL FILE AND SAMPLE DATA FILE IN AMPL TO FIND OPTIMAL ALLOCATION OF SIDE TWO WEAPONS

Appendix A
MODEL FILE AND SAMPLE DATA FILE IN AMPL TO FIND OPTIMAL ALLOCATION OF SIDE TWO WEAPONS

Model file:

```
set target;                          # targets being attacked
set weapon;                          # weapons used to attack
param payout {weapon, target} >= 0;           # payout for each weapon at each target
param supply {weapon} >= 0;                   # number of each type of weapon
param goal {target} >= 0;                     # number of each type of target
var allocation {weapon, target} >= 0;         # number of weapons allocated to each target
maximize total_payout:
        sum{i in weapon, j in target} payout[i,j] * allocation[i,j];
         # choose the allocation that maximizes the total payout
subject to Demand{j in target}:
        sum{i in weapon} allocation[i,j] <= 2*goal[j];
         # assigned warheads must be less than total number of targets
subject to Supply{i in weapon}:
        sum{j in target} allocation[i,j] <= supply[i];
         # assigned warheads must be less than the total number of warheads
subject to Destroyed{j in target}:
        sum{i in weapon} payout[i,j] * allocation[i,j] <= goal[j];
         # destroyed targets does not exceed total targets
```

Sample data file:

```
param payout:
         t1    t2   :=
        w1    0.9  0.7
        w2    0.8  0.3      ;
         # columns represent targets, rows represent weapons
param: weapon: supply :=                      # number of weapons available
        w1    100
        w2    100 ;
param: target: goal :=                        # number of targets available
        t1    100
        t2    100 ;
Output from AMPL:
Objective Function = 150
Allocation:
         t1    t2
        w1    0    100
        w2    100 0
```

A-1

(This page is intentionally blank.)

Appendix B

CALCULATION OF SIDE ONE PAYOUT MATRIX

Appendix B
CALCULATION OF SIDE ONE PAYOUT MATRIX

λ: **0.3**

Pk Values (Calculated from DE and RD)

Side One Weapons	Side Two Force and Value Targets					
	w1	w2	w3	t1	t2	t3
w1	0.35	0.40	0.50	0.75	0.85	0.50
w2	0.35	0.28	0.60	0.75	0.85	0.50
w3	0.26	0.42	0.10	0.62	0.77	0.50

Side Two Weapons	Side One Value Targets			Calc of - ΔF	- ΔF
	t1	t2	t3		
w1	0.90	0.70	0.50	0.9 - (0.9 - 0.7)	0.70
w2	0.70	0.30	0.50	0.7 - (0.9 - 0.7)	0.50
w3	0.40	0.40	0.80	0.8 - (0.8 - 0.8)	0.80

Calculation of "Payout Matrix" Values

Side One Weapons	Side Two Force and Value Targets					
	w1	w2	w3	t1	t2	t3
w1	0.35*0.70	0.40*0.50	0.50*0.80	0.3*0.75	0.3*0.85	0.3*0.50
w2	0.35*0.70	0.28*0.50	0.60*0.80	0.3*0.75	0.3*0.85	0.3*0.50
w3	0.26*0.70	0.42*0.50	0.10*0.80	0.3*0.62	0.3*0.77	0.3*0.50

"Payout Matrix" Values

Side One Weapons	Side Two Force and Value Targets					
	w1	w2	w3	t1	t2	t3
w1	0.25	0.20	0.40	0.23	0.26	0.15
w2	0.25	0.14	0.48	0.23	0.26	0.15
w3	0.18	0.21	0.08	0.19	0.23	0.15

(This page is intentionally blank.)

Appendix C

ONEGULP MODEL AND SAMPLE DATA FILE FOR THE OPTIMAL ALLOCATION OF SIDE ONE WEAPONS

Appendix C
ONEGULP MODEL AND SAMPLE DATA FILE FOR THE OPTIMAL ALLOCATION OF SIDE ONE WEAPONS

Model File:

```
set weapon1;                                  # warheads for each weapon type for side 1
set side2;                                    # force and value targets for side 2

param payoutw1vs2 {weapon1, side2} >= 0;      # payout for each weapon vs each target
param pkw1vs2 {weapon1, side2} >= 0;          # pk values for each weapon-target combo
param supplyw1 {weapon1} >= 0;                # number of weapons available for side 1
param targets2 {side2} >= 0;                  # number of force and value targets on side 2
param bases2 {side2} >= 0;                    # number of bases of each target type for side 2
param targets_per_base {side2} >= 0;          # number of WH or value targets at each base
        check {j in side2}: targets2[j] = bases2[j] * targets_per_base[j];

var allocationw1vs2 {weapon1, side2} >= 0;
         # allocation of side 1 weapons to side 2 force and value targets

maximize total_payout:
        sum{i in weapon1, j in side2} payoutw1vs2[i,j] * targets_per_base[j] *
allocationw1vs2[i,j];
         # maximizes the damage avoided and the damage inflicted

subject to Demand{j in side2}:
        sum{i in weapon1} allocationw1vs2[i,j] <= 2*bases2[j];
         # no more than 2 warheads (WH) allocated per base
subject to Supply{i in weapon1}:
        sum{j in side2} allocationw1vs2[i,j] <= supplyw1[i];
         # WH fired does not exceed WH available
subject to Destroyed{j in side2}:
        sum{i in weapon1} pkw1vs2[i,j] * allocationw1vs2[i,j] * targets_per_base[j] <=
targets2[j];
         # destroyed targets does not exceed total targets
```

Data File:

```
param payoutw1vs2:
        w1    w2      w3    w4    w5    t1    t2    t3    :=
w1      0.31  0.31    0.00  0.72  0.72  0.04  0.25  0.00
w2      0.31  0.31    0.00  0.72  0.72  0.04  0.25  0.00
w3      0.23  0.23    0.00  0.72  0.72  0.03  0.25  0.00
w4      0.23  0.23    0.00  0.72  0.72  0.03  0.25  0.00
w5      0.00  0.00    0.00  0.72  0.72  0.06  0.25  0.00    ;
         # payout matrix (calculated by method in Appendix B)
         # columns represent Side Two targets, rows represent Side One weapons
```

```
param pkw1vs2:
              w1     w2       w3     w4     w5     t1     t2     t3     :=
w1            0.37   0.37     0.00   0.85   0.85   0.12   0.85   0.00
w2            0.37   0.37     0.00   0.85   0.85   0.12   0.85   0.00
w3            0.27   0.27     0.00   0.85   0.85   0.09   0.85   0.00
w4            0.27   0.27     0.00   0.85   0.85   0.09   0.85   0.00
w5            0.00   0.00     0.00   0.85   0.85   0.19   0.85   0.00   ;
               # Pks for Side One weapons against Side Two targets

param: weapon1: supplyw1 :=          # number of Side One WH available
              w1     500
              w2     500
              w3     500
              w4     500
              w5     1000     ;

param: side2: targets2 :=            # number of Side Two targets
              w1     500
              w2     500
              w3     500
              w4     500
              w5     1000
              t1     1500
              t2     1500
              t3     0    ;

param: bases2 :=                     # number of bases for each target type
              w1     500
              w2     500
              w3     1
              w4     2
              w5     2
              t1     1500
              t2     1500
              t3     0    ;

param: targets_per_base :=           # number of targets per base
              w1     1
              w2     1
              w3     500
              w4     250
              w5     500
              t1     1
              t2     1
              t3     0    ;
```

Appendix D

GLOSSARY

Appendix D
GLOSSARY

AMPL	A Mathematical Programming Language
ASCO	Advanced Systems and Concepts Office
DE	damage expectancy
DTRA	Defense Threat Reduction Agency
IDA	Institute for Defense Analyses
kB	kilo bytes
LANL	Los Alamos National Laboratory
MESA	Multiple Exchange of Strategic Arsenals
MHz	megahertz
ONEGULP	Optimized Nuclear Exchange Games Using Linear Programming
P_k	probability of kill
RD	required damage

(This page is intentionally blank.)

Appendix E

DISTRIBUTION LIST FOR IDA DOCUMENT D-2743

Appendix E
DISTRIBUTION LIST FOR IDA DOCUMENT D-2743

Office of the Secretary of Defense	No. of Copies
Defense Threat Reduction Agency/ACS 8725 John J. Kingman Road, MS 6201 Fort Belvoir, VA 22060-6201 ATTN: Larry Sanders	1
National Laboratories	
Los Alamos National Laboratory P.O. Box 1663 Physics Division Los Alamos, NM 87545 ATTN: Dennis Powe11	1
Other Organizations	
Ms. Catherine D. Farmer Box 0678 6 Frist Campus Center Princeton, NJ 08544	1
Dr. Larry Farmer Box 50 Middle Tennessee State University Murfreesboro, TN 37132	1
Professor Warren Powell E405 Engineering Quardrangle Princeton University Princeton, NJ 08544	1
Dr. Ivan Oelrich Federation of American Scientists Suite 209 1717 K Street, NW Washington, DC 20036	1

Other Organizations (Continued)	No of Copies
Dr. Jerome Bracken 5 Magnolia Parkway Chevy Chase, MD 20815	1
Defense Technical Information Center 8725 John J. Kingman Road, Suite 0944 Fort Belvoir, VA 22060-6218	1
Institute for Defense Analyses 4850 Mark Center Drive Alexandria, VA 22311-1882	6
Total Distribution	**14**

REPORT DOCUMENTATION PAGE		Form Approved OMB No. 0704-0188

1. REPORT DATE (DD-MM-YY)	2. REPORT TYPE	3. DATES COVERED (FROM – TO)
August 2002	Final	June 2002–August 2002

4. TITLE AND SUBTITLE	
A Linear Programming Approach to Complex Games: An Application to Nuclear Exchange Models	5A. CONTRACT NO. DASW01-98-C-0067
	5B. GRANT NO.
	5C. PROGRAM ELEMENT NO(S).
6. AUTHOR(S) I.C. Oelrich, C.D. Farmer	5D. PROJECT NO.
	5E. TASK NO. CRP 1082
	5F. WORK UNIT NO.
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Institute for Defense Analyses 4850 Mark Center Drive Alexandria, VA 22311-1882	8. PERFORMING ORGANIZATION REPORT NO. IDA Document D-2743
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) IDA Contracting Officer's Representative 4850 Mark Center Drive Alexandria, VA 22311-1882	10. SPONSOR'S / MONITOR'S ACRONYM(S)
	11. SPONSOR'S / MONITOR'S REPORT NO(S).

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This paper reports an analytical solution to a two-sided nuclear exchange using simple linear programs. This solution is proposed as an alternative to MESA, a model developed at Los Alamos National Laboratory that uses an optimizing search algorithm. Like the MESA model, the exchange is cast in terms of game theory, using linear approximations and an optimal allocation defined by a user-specified objective function. Solutions are better using linear programs instead of optimizing searches, and the solutions are many times faster.

15. SUBJECT TERMS

game theory, nuclear exchange model, MESA, Multiple Exchange of Strategic Arsenals, Los Alamos National Laboratory

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NO. OF PAGES	19A. NAME OF RESPONSIBLE PERSON
			Unlimited	42	Mr. Jay S. Minsky
A. REPORT	B. ABSTRACT	C. THIS PAGE			19B. TELEPHONE NUMBER (INCLUDE AREA CODE)
Unclassified	Unclassified	Unclassified			(703) 845-2202